#### ****项目创建****

创建一个文件夹，进入该目录，在该目录下打开一个终端，

执行npm init。根据提示输入内容，也可以直接按回车键跳过。执行完后目录中会多一个package.json文件

#### ****修改package.json****

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"start": "node\_modules/.bin/webpack"

},

#### ****安装react, react-dom, webpack****

在项目根目录下执行下面命令，其中--save的含义是项目上线运行所需的包，即生产环境（--save-dev是开发环境所需的包）。

npm install react react-dom --save  
npm install webpack webpack-cli --save-dev

#### ****引入Babel库进行转码****

npm install @babel/core babel-loader @babel/preset-env @babel/preset-react --save-dev

配置Babel，创建一个名为.babelrc 文件在根项目文件夹中：

{

  "presets": ["@babel/preset-env", "@babel/preset-react"]

}

**安装html-webpack-plugin插件**

npm install html-webpack-plugin --save-dev

**webpack.config.js文件配置（**根项目文件夹**）**

const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {

devtool: 'eval-source-map',

entry: \_\_dirname + "/src/App.jsx",

output: {

path: \_\_dirname + "/build",

filename: "index.js"

},

module: {

rules: [

{

test: /\.(js|jsx)$/,

use: {

loader: 'babel-loader'

},

exclude: /node\_modules/

}

]

},

plugins: [

new HtmlWebpackPlugin({

filename: \_\_dirname + '/build/Index.html'

})

]

}

#### ****在src文件夹下创建App.jsx文件****

import React, { Component } from 'react';

import ReactDOM from 'react-dom';

class App extends React.Component{

render(){

return (<h1>holle world</h1>);

}

}

ReactDOM.render(

<App />,

document.body);

#### ****运行如下命令****

npm start